

**RESPOSTA:** O código em Python ocupou menos linhas devido a certos recursos da linguagem, como por exemplo não ser necessário declarar o tipo de variáveis, nem declarar variáveis primeiro para só depois poder utilizá-las para armazenar a entrada do usuário (faz tudo numa linha só com `opcao = input(f"--> Digite o numero da opcao desejada: ")`) e nem precisar importar bibliotecas/arquivos de cabeçalho para poder lidar com funções de saída (como o `stdio.h`), nem definir os tipos de entrada e dos valores a serem exibidos na saída, dentre outras etapas que são necessárias em C. Assim sendo, o exemplo em Python ficou muito mais legível se comparado à mesma versão em C, mas isso em termos de legibilidade e menor quantidade de linhas. Entretanto, devido à existência do `switch/case` em C, o código em C ficou mais claro quanto ao propósito da estrutura condicional múltipla, se comparado com o monte de `if/else` aninhados em Python, que ficou mais parecido com uma “gambiarra”. Então, por mais que o exemplo em Python tenha ficado mais enxuto, a falta de uma estrutura condicional de múltipla escolha e a implementação com condicionais aninhadas deixa a lógica de avaliação condicional múltipla mais difícil, enquanto o `switch/case` do C deixa esta lógica mais clara; portanto, concluo que a implementação mais fácil foi em C devido à presença do `switch/case`. Entretanto, vale a pena destacar que nas últimas versões do Python também há uma estrutura de condicionais múltiplas nativa da linguagem, o `match/case`, que é semelhante ao `switch/case` do C e extremamente poderosa, embora não seja tão utilizada no desenvolvimento Python ainda. Então concluo que a implementação em C deixou a lógica mais clara, mas o código em Python foi mais breve e enxuto, embora tenha “perdido” para C por causa das múltiplas condicionais aninhadas que podem se parecer um pouco como uma “gambiarra”, de certa forma.